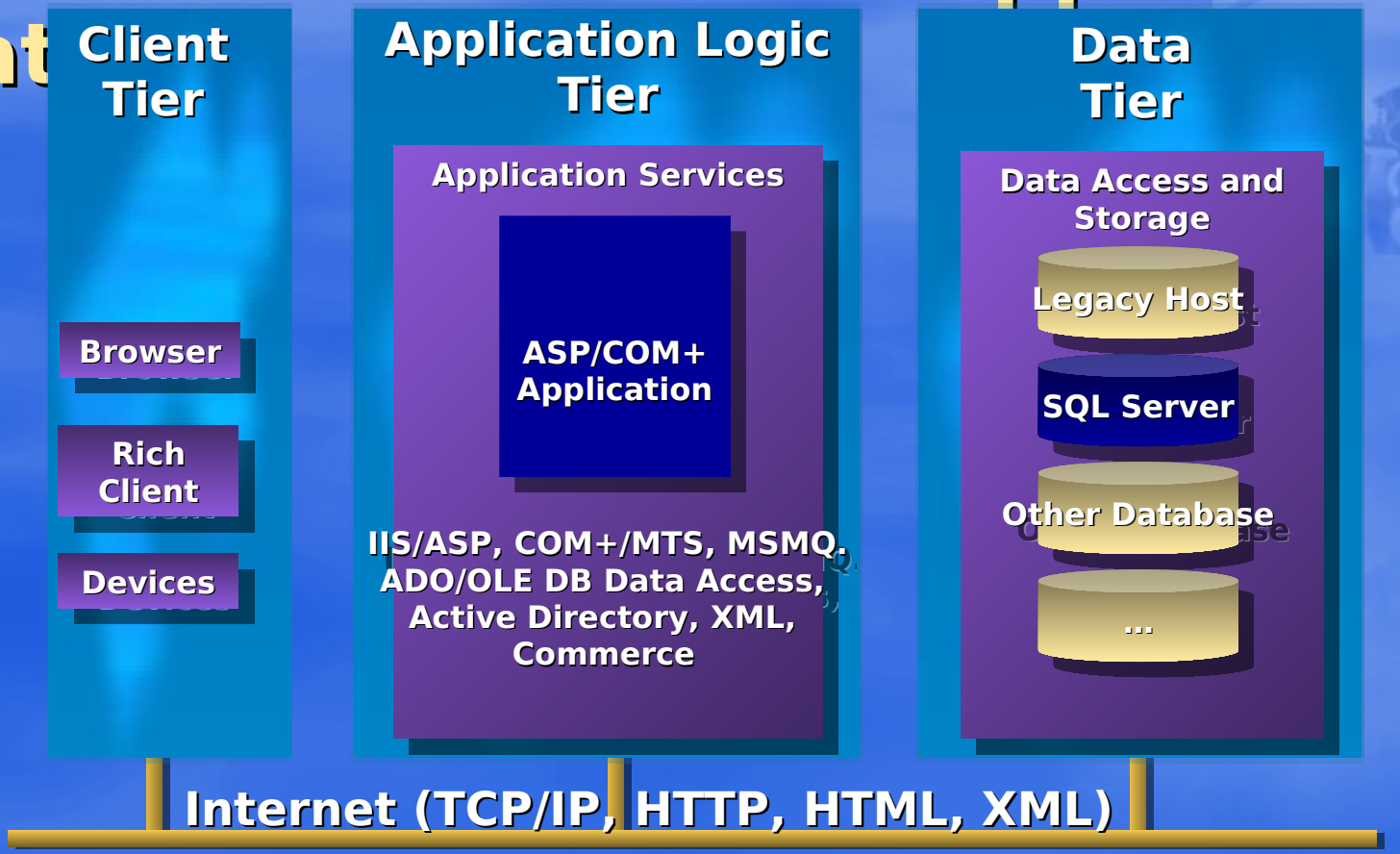
SQL Server 2000 Performance Improvements

Damien Lindauer
Group Manager
SQL Server Performance Engineering
Microsoft Corporation


POWER UP
Windows DNA 2000
Readiness Conference
featuring SQL Server 2000

Windows DNA 2000

Next Generation Web Application Plat



Microsoft SQL Server 2000

Microsoft Application Center 2000

Microsoft Windows 2000 Server

Microsoft Commerce Server 2000

Microsoft Host Integration Server 2000

Microsoft BizTalk Server 2000

SQL Server 7.0
Self Configuring, Managing And Tuning

- Dynamic memory/lock/object management
- Increased scan and random I/O throughput
- Automatic intra-query parallelism
- Integrated, intelligent Readahead
- Automatic statistics creation/update
- Advance query optimization techniques
- Automated index tuning
- Query plan caching

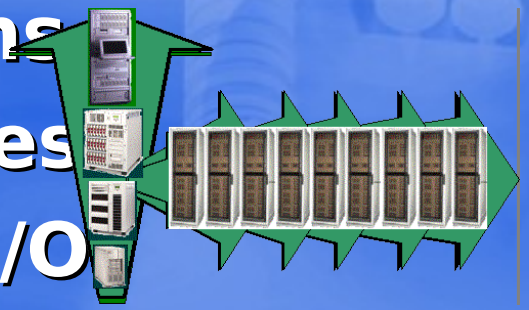
SQL Server 2000
Performance Objective

SQL Server 2000

Address current customer requirements and future performance trends by extending and enhancing the advanced and flexible architecture of SQL Server 7.0

Current Customer Requirements And Future Performance Trends

- Increasing user populations
- Increasing transaction rates
- Increasing database disk I/O
- Increasing data and query complexity
- Increasing application network I/O
- Stricter response time requirements
- Need to “Scale-Up” on SMP platforms



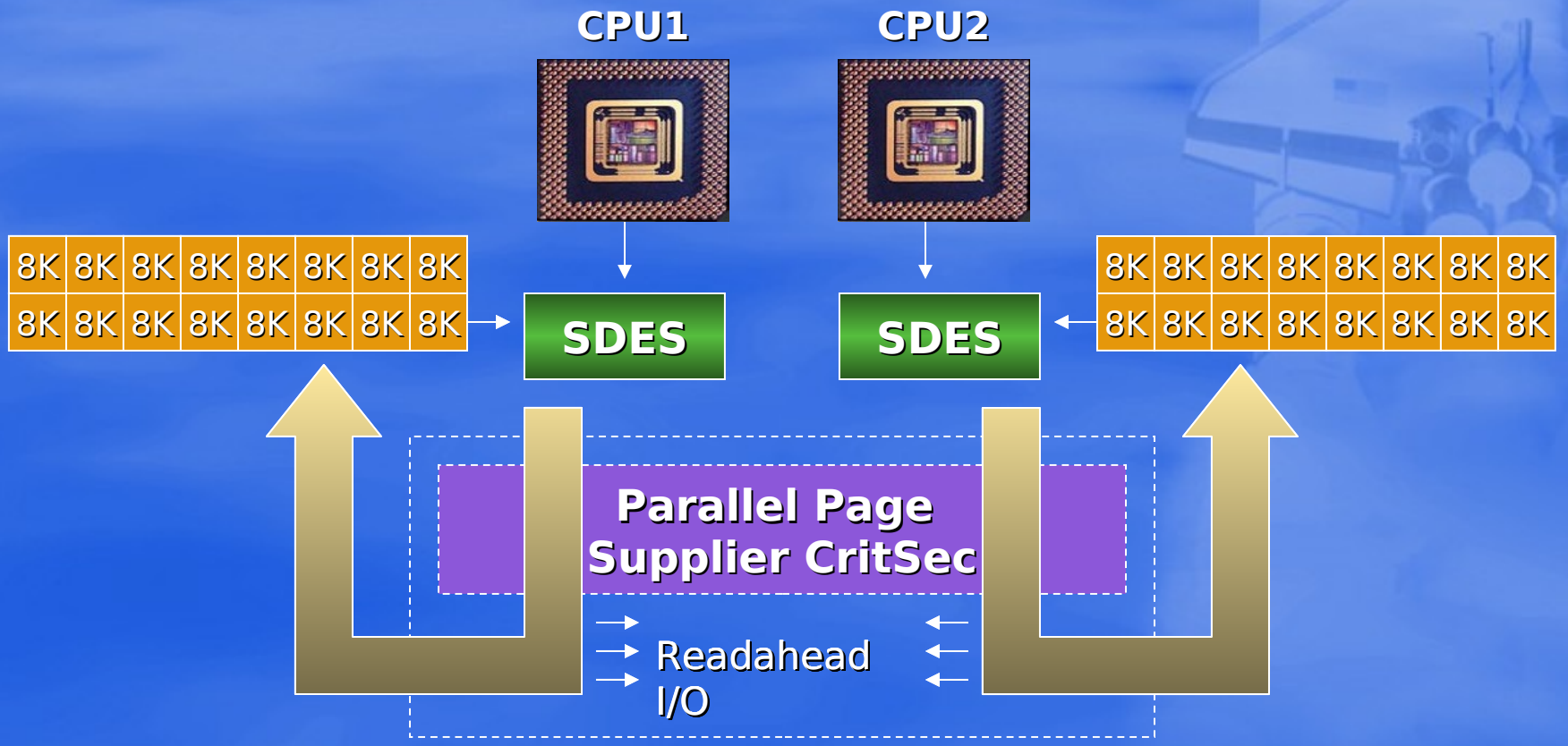
Performance Challenge
Increasing Disk I/O

- Drive larger/faster I/O subsystems
- More efficient use of large memory
- Support for TB+ size databases
- Optimize for OLTP I/O profiles
- Optimize for DSS I/O profiles
- Handle text/image data efficiently

SQL Server 2000 Solutions

- Improved parallel scan processing
- Merry-go-round scans
- More scalable Readahead
- Dynamic asynchronous I/O
- Large memory support (AWE)
- Scalable checkpoint processing
- Efficient “Text in Row” format

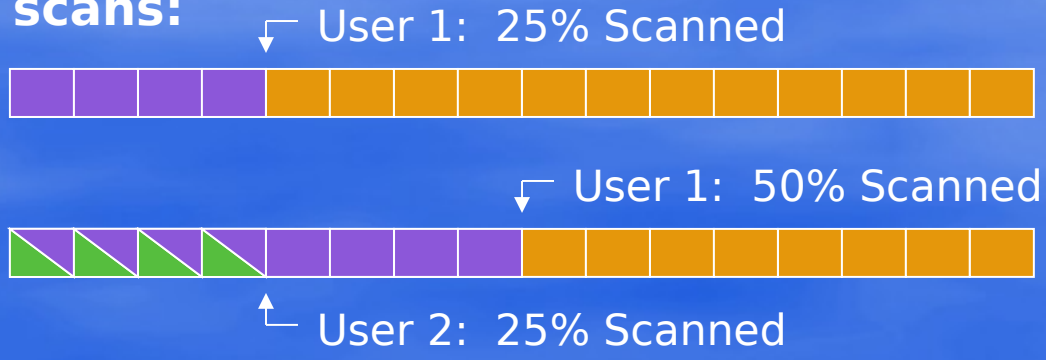
Improved Parallel Scan



- Less time spent in parallel page supplier critical section
- Readahead I/O now issued outside the critical section
- Architecture feeds 16 pages at a time (fewer calls to PPS)

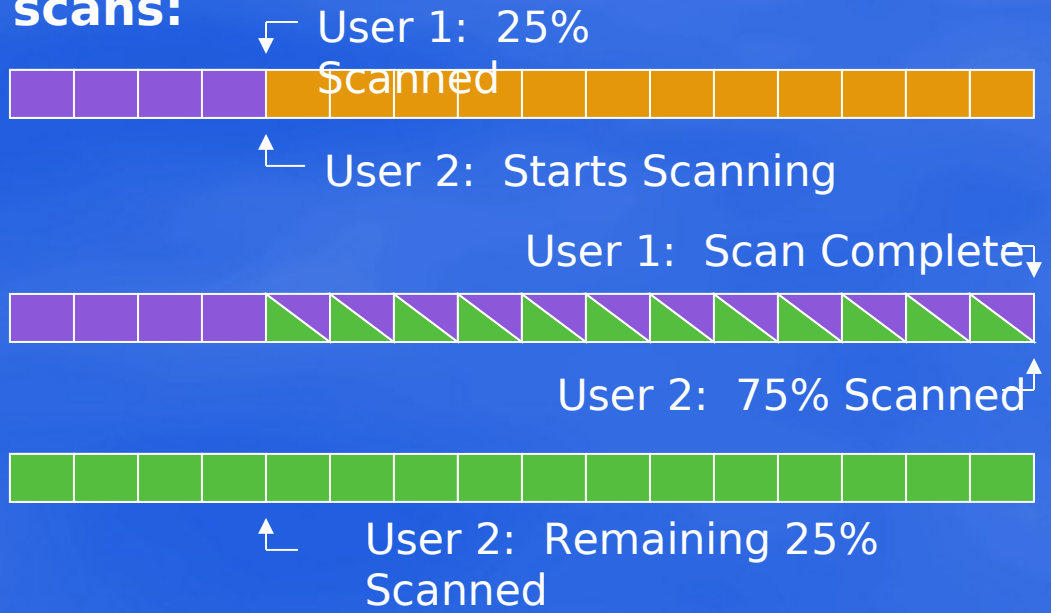
Merry-Go-Round Scans

Without Merry-go-round scans:



Can result in disk thrashing during large table scans!

With Merry-go-round scans:



- **Reduces disk thrashing**
- **Improves cache behavior**
- **Affects unordered scans**
- **No synchronization**

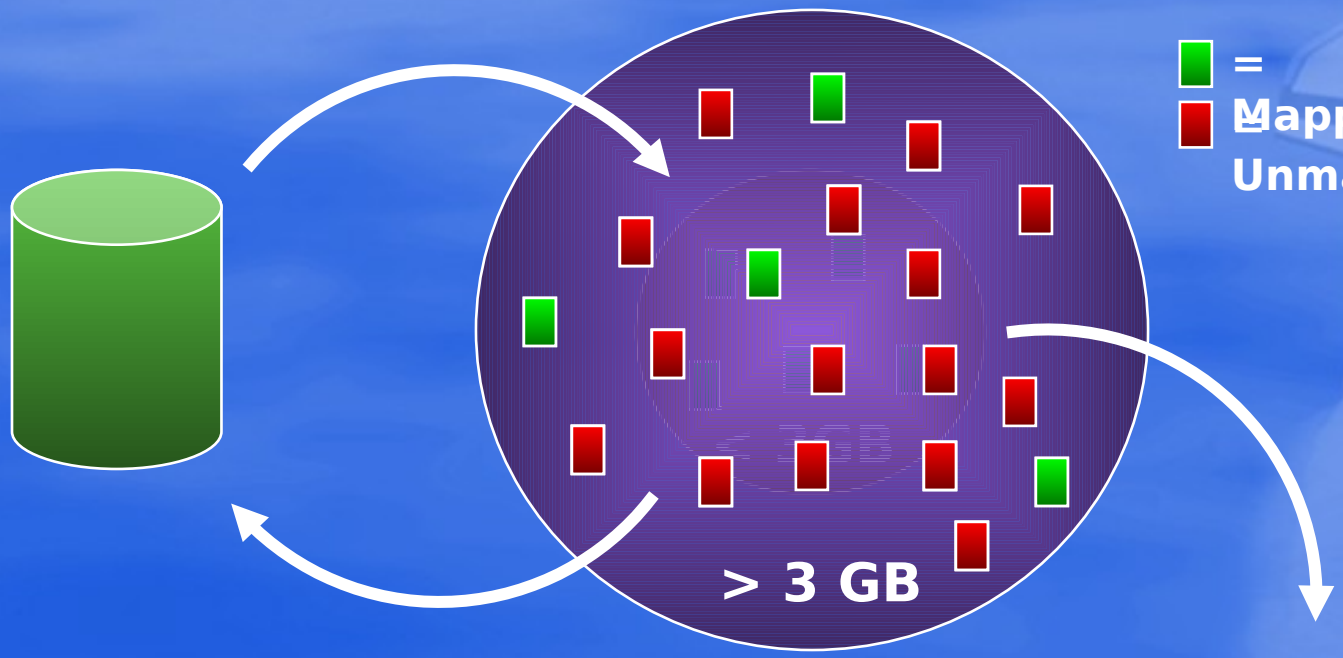
More Scalable Readahead

- Deeper Readahead processing
  - Uses up to 1% of available memory or 2000 buffers (whichever is smaller)
- Benefits:
  - Better utilization of large I/O subsystems
  - Improved scaling on SMP systems
- Bonus:
  - Integrated Readahead during

Dynamic Asynchronous I/O

- Max async I/O no longer configured statically by parameter
- Async I/O now increases with demand
- Affects both read and write I/O
- Algorithm automatically paces itself
- Designed to allow optimal saturation of I/O subsystems

Large Memory Support



- Virtual address space still limited to 3GB
- AWE allows 36bit physical address space via PAE (up to 64gb)
- Pages mapped/unmapped between virtual and physical
- Maps are magnitude less costly than copies
- Enabled via “awe enabled” parameter (static non-paging)

Scalable Checkpoint

Chkpt starts

Results in Fewer Checkpoint I/Os on Busy Systems!

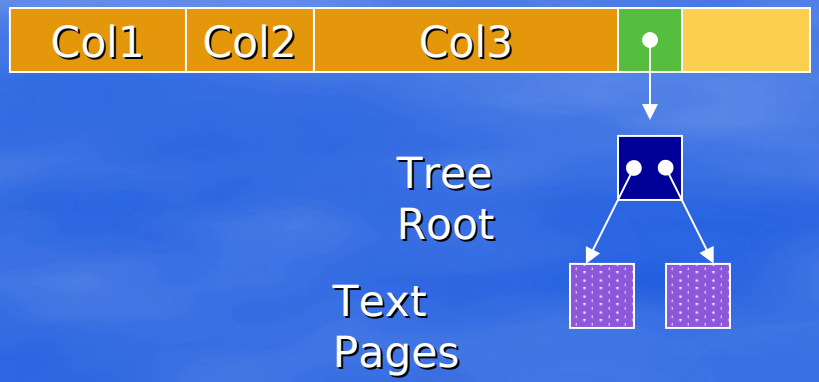
Chkpt continues

...while Lazywriter flushes pages

Chkpt completes

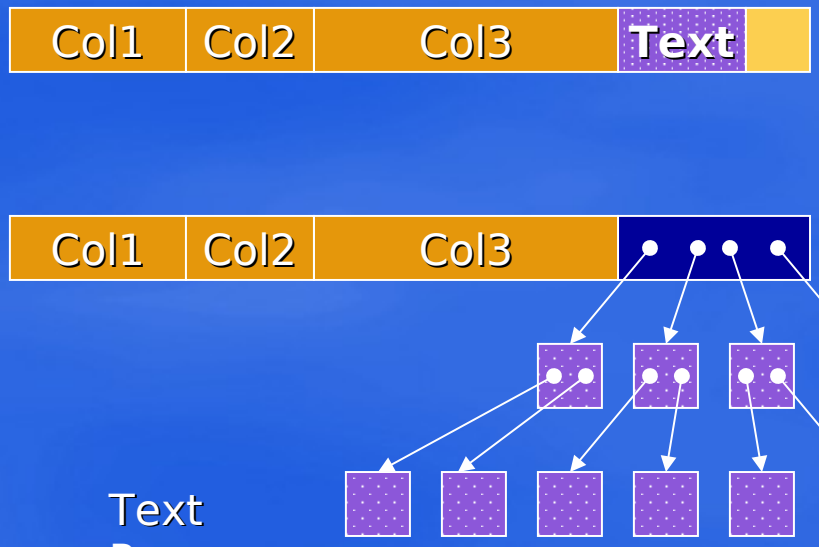
Text In Row Format

SQL 7.0 Text Format:



- Text stored “off row” in tree
- Text pointer jump to root

Text In Row Format:



- Text stored “in row” if it fits
- Otherwise, root stored “in row”
- Reduced I/O for text retrieval
- Enabled/disabled by table option
- Optional “inline limit” (256 default)
- Increased I/O for non-text scans

Performance Challenge
Processor and SMP Scale-Up

- More efficient use of CPU resources
- Reduction of critical code paths
- Elimination of unnecessary processing
- Improve overall SMP scaling
- Improve processor affinity
- Reduce cross processor cache traffic/contention

SQL Server 2000 Solutions

- Server side cursor caching
- Parameter caching
- Enhanced intra-query parallelism
- Full parallel index creation
- Partitioned free buffer lists

Server Side Cursor Caching

- Cursor compile plan
  - Saves allocation of object
- Cursor execution plan
  - Representation of cursor is cached
- Benefits:
  - Workloads dominated by cursor operations
  - 10-15% reduction in CPU consumption for cursor operations
- No application changes

Parameter Caching

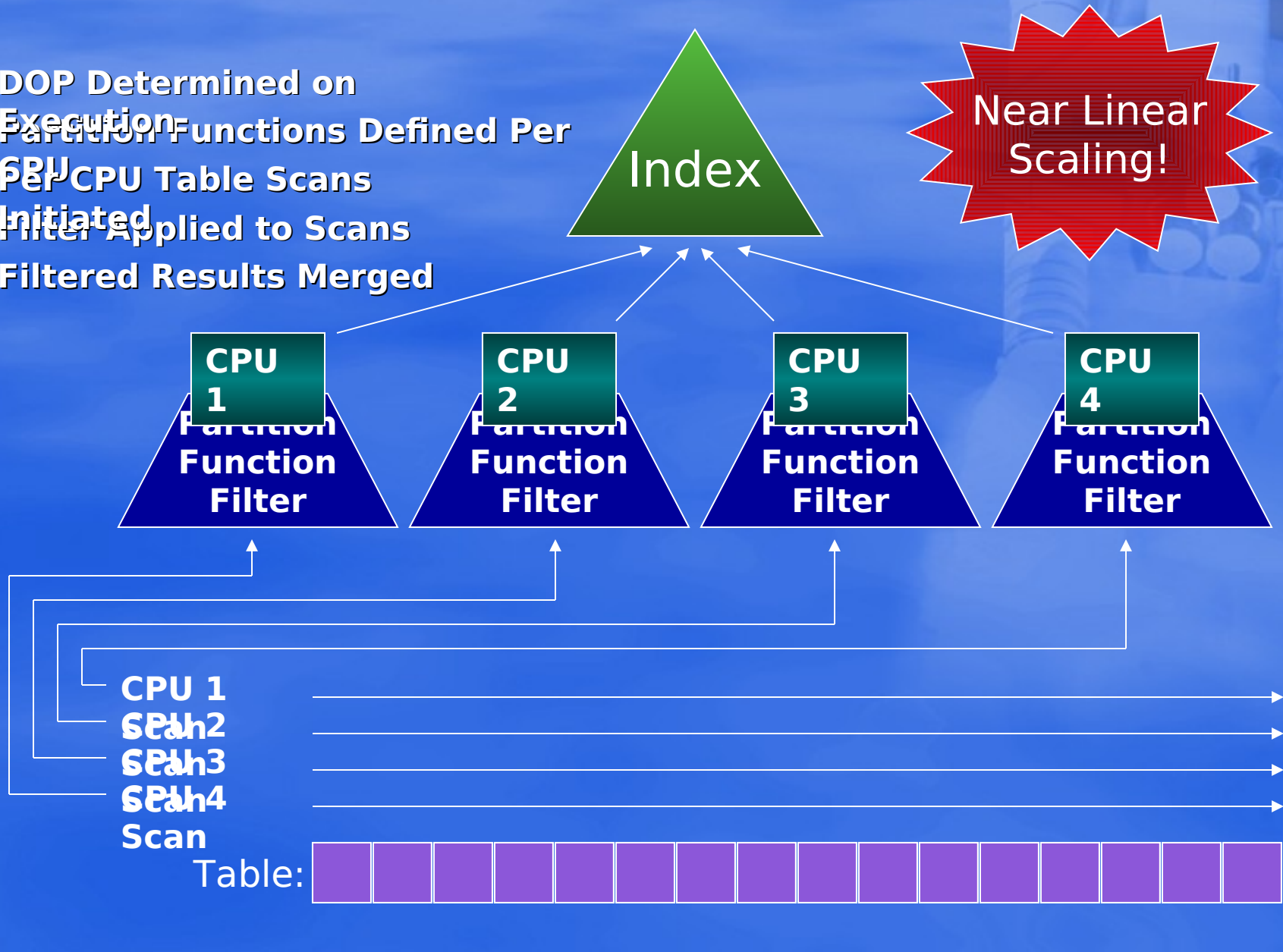
- Type checks and default parameter values cached on first execution
- Subsequent executions use cached information and default values
- Re-cached if plan recompile/schema change
- Benefits:
  - Heavy stored procedures use with many parameters/defaults
  - 6% improvement over SQL Server 7.0 on SAP SD workload
- No application changes necessary (if using prepare/execute to call stored procedure)

Intra-Query Parallelism

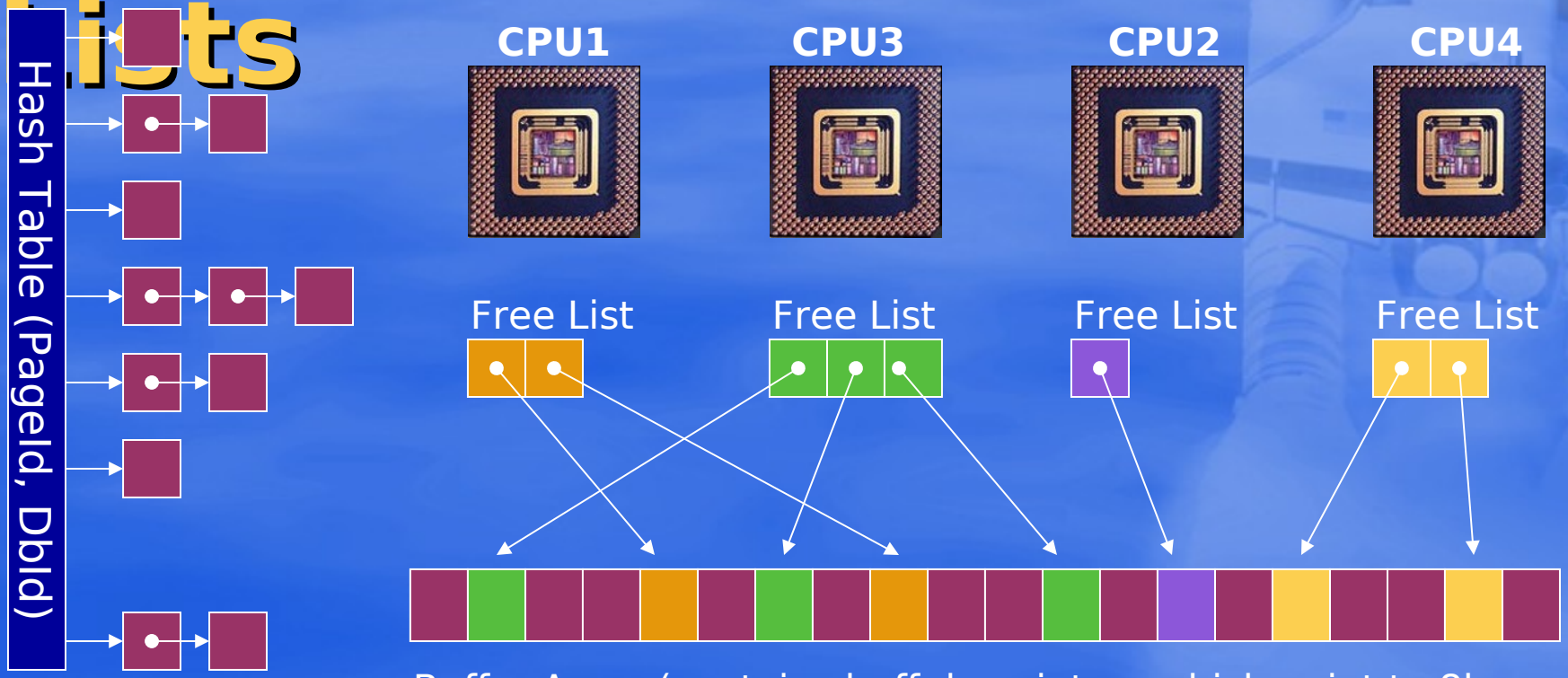
- Integrated bitmap filtering
- Multi-threaded access to spools
- Reduced parallel sort stalls
- Elapsed time costing
- Insert/Update/Delete integration
- Results:
  - Improved TPC-H DSS workload by 40% over SQL Server 7.0

Parallel Index Creation

- DOP Determined on Execution
- Partition Functions Defined Per CPU
- Per CPU Table Scans Initiated
- Filter Applied to Scans
- Filtered Results Merged



Partitioned Free Buffer Lists



- **Results in Free List per CPU**
- **Enhances cache locality of free list descriptors and buffers**
- **Distributed Lazywriting balances population of free lists**
- **Improves Scale-Up on SMP platforms**
- **Architecture in place for further NUMAish enhancements**

Performance Challenge
Increasing Network I/O

- Larger and more complex app architectures (2-tier, N-tier, hybrid)
- App tier network traffic increasing (eCommerce driven)
- Data tier network traffic increasing
- Stricter response time requirements

SQL Server 2000 Solutions

- Prepare/unprepare caching to reduce network roundtrips
- Metadata caching to reduce data on wire
- Support for high-speed System Area Networks (SAN)

Prepare/Unprepare Caching

- Prepare/unprepare cached on client
  - Prepare piggybacked on first exec
  - Unprepare piggybacked on next prepare
- Removes N+2 roundtrip scenario
- Overall benefit inversely related to increase in execs of prepared plan

Metadata Caching

- Metadata cached on client between subsequent calls on connection
- Coordinated between client/server
  - Metadata only sent if schema change
- Data reduction inversely related to # of columns (~10 bytes per column)
- Bonus: Saves CPU required to process metadata on both client and server!
  - Reduces cost of select with > 50 columns by over 10%

SAN Support

- Socket Netlib supports WSD (DC only)
- Native VI Netlibs (Servernet and Giganet)
- High bandwidth, low latency, reliable
- Bonus:
  - Low system CPU overhead
- Benefits:
  - Network intensive client/server apps
  - Scale-Out data tier interconnects
  - Reduced kernel time by 1/3 in SAP

Performance Challenge
Data and Query Complexity

- Efficient query processing of ever increasing data set sizes
- Increasing query complexity for business intelligence and data warehousing systems
- Understanding of the “nature of the data”
- Efficient execution of multi-user workloads with mixed query complexities

SQL Server 2000 Solutions

- Improved statistics collection and representation
- Even more efficient plans
- Advanced index enhancements
- Improved multi-user robustness

Improved Statistics

- Similar Bucket Merge Algorithm (SBM) builds maxdiff histogram for statistics
  - Dynamic bucket boundaries based on variances
  - Effectively handles high low and high data skew
  - Captures density and distribution in one pass
  - Improves cardinality estimates by creating more granular statistics
- Optimizer still understands SQL 7.0 statistics
  - Necessary in case where database is upgraded
  - Will update to new format automatically

More Efficient Plans

- More granular and accurate statistics means improved cardinality estimates
  - Better understanding of “nature of data”
  - Higher quality input for optimizer
- Modular and flexible QP design has allowed “fine tuning” for unique and complex queries
- Even more “trivial plan” cases supported

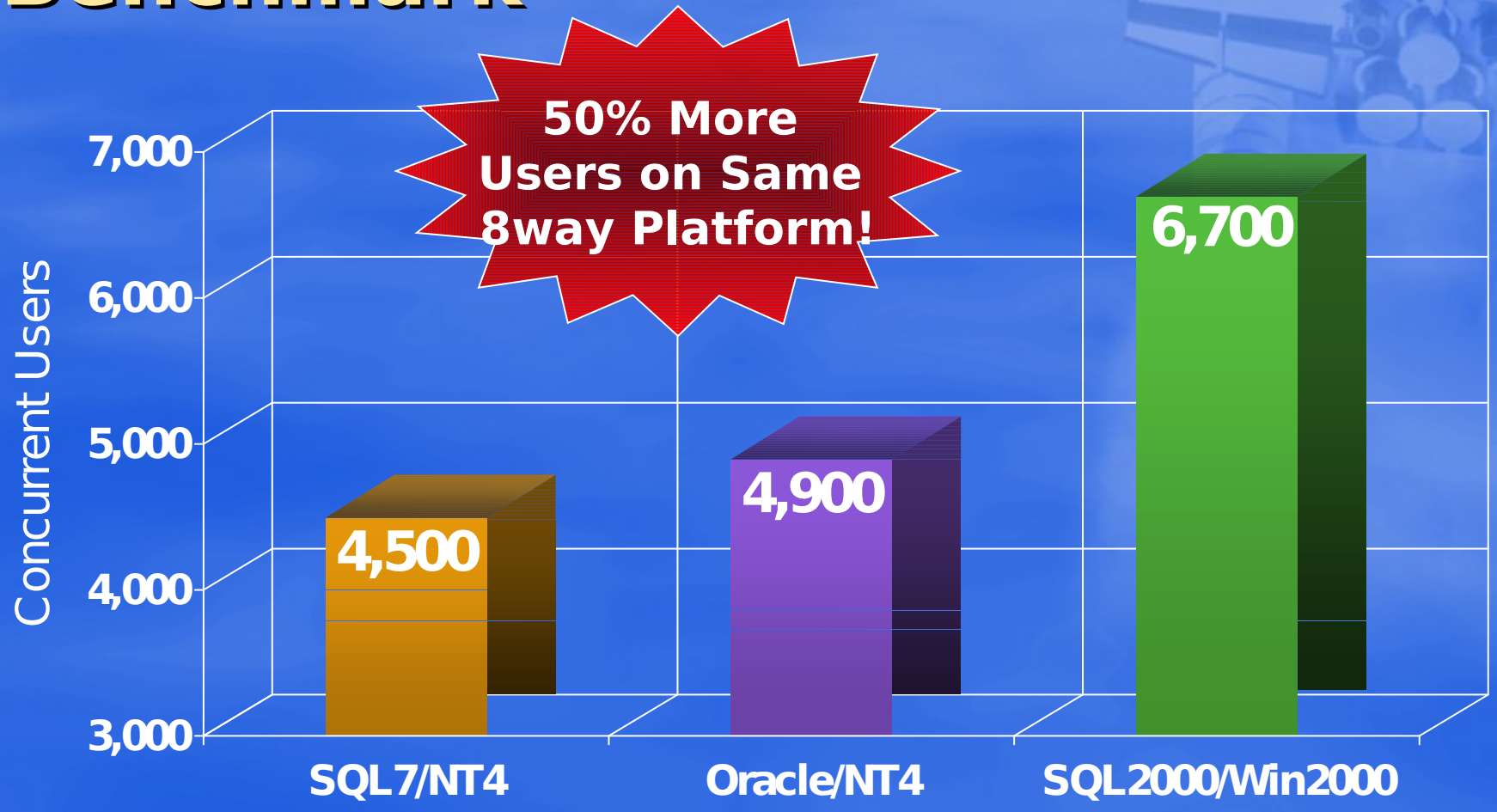
Indexed Views

- Index on view may contain join, aggregation, or combination
- Once created, contents of view persist
- Optimizer may use index even if not directly referenced by query
- Benefits:
  - Decision Support Systems
  - Data Warehouses / DataMarts
  - Business Intellegence Systems
- Integration with OLAP Services
- Consider cost of index

Multi-User Robustness

- Complex or long running queries require resources (memory/CPU)
- SQL Server 2000 implements more intelligent scheduling of queries
- Prior to execution, QP will examine the “cost” of query relative to other current query activity
- QP may choose to delay

Scale-Up Results!
SAP SD Industry Standard Benchmark



Performance Challenge
Data Tier Scale-Out

- Eliminate single server bottleneck
- Provide scalable performance/capacity
- Application transparency to tables
- Intelligent distributed query
- Distributed transaction coordination
- Integration with cluster

SQL Server 2000 Solutions

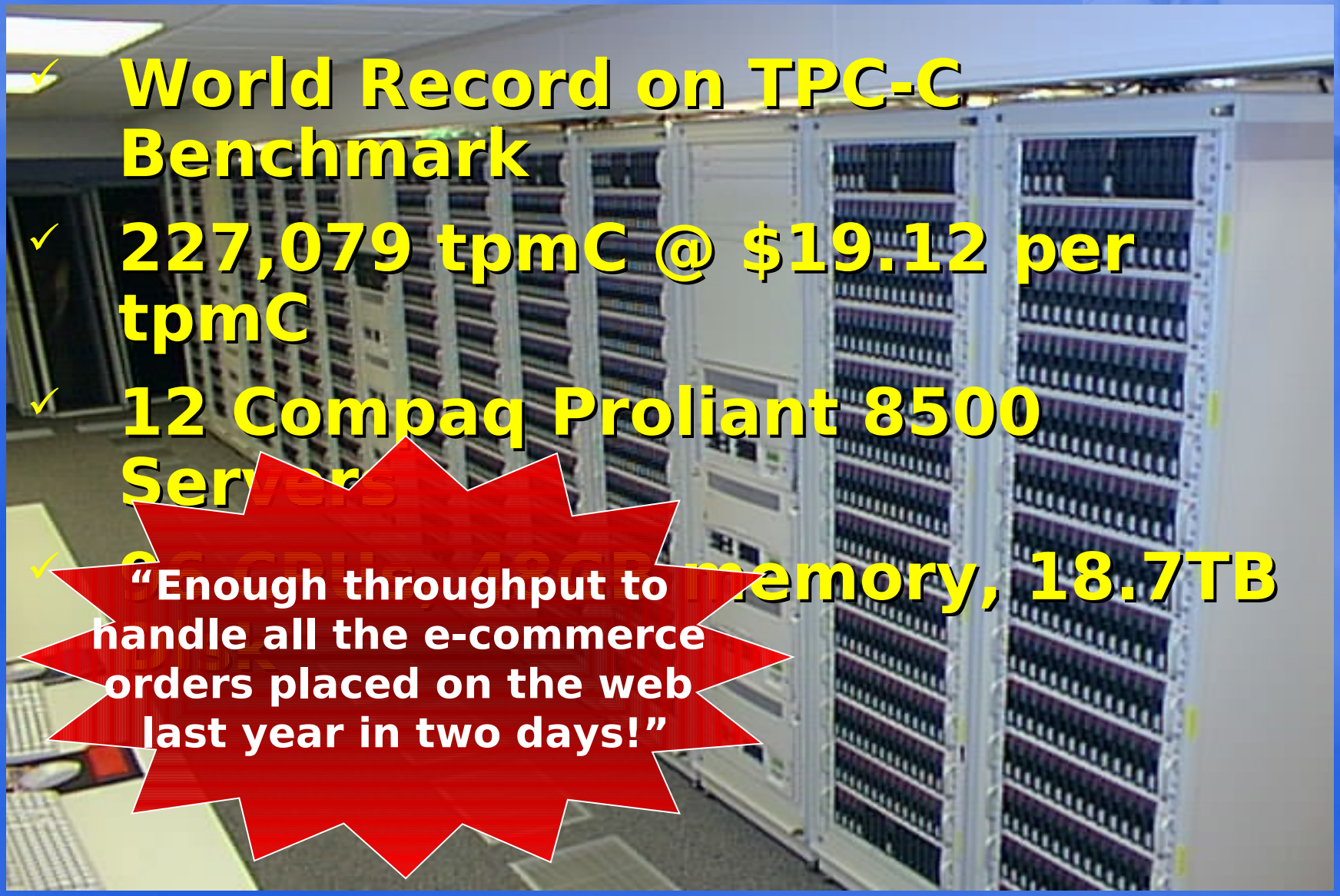
Building blocks for Scale-Out

- Distributed Transaction Coordinator
- OLEDB/Distributed Query
- Distributed Partioned Views
- Instead Of Triggers
- System Area Network support
- Windows Cluster Services
- IIS/ASP/ADO/COM+
- Windows 2000 Cluster Services

Distributed Partitioned View

- UNION ALL view over local table and one or more remote “member” tables
- Member tables have identical structure with unique partitioning column
- Disjoint partitioning intervals enforced with CHECK constraints
- Participating member servers use OLEDB linked server support
- DTC maintains transactional integrity
- QP optimizes distributed plans

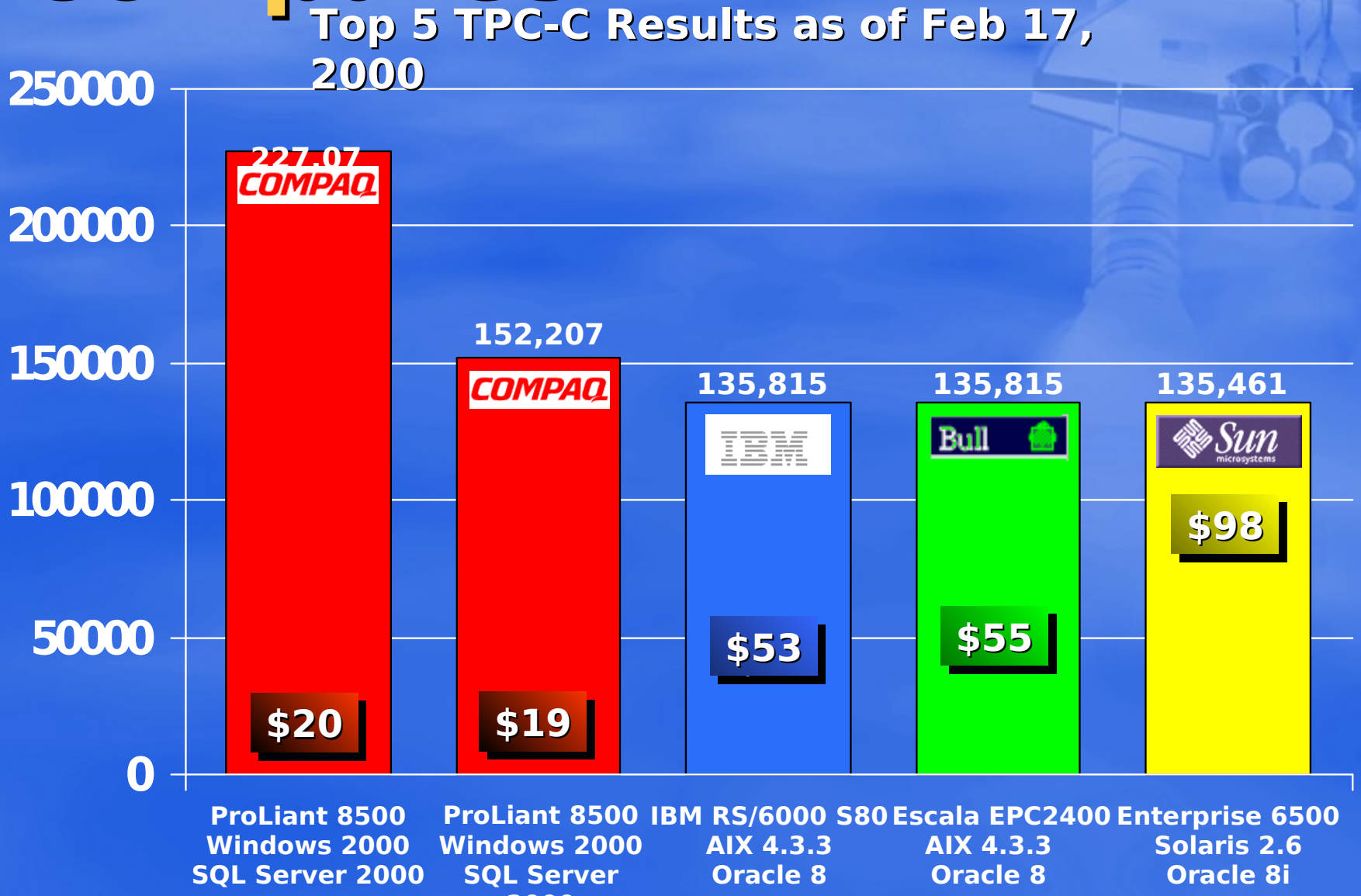
Scale-Out Results!



- ✓ World Record on TPC-C Benchmark
- ✓ 227,079 tpmC @ $19.12 per tpmC
- ✓ 12 Compaq Proliant 8500 Servers
- ✓ [illegible] emory, 18.7TB

"Enough throughput to handle all the e-commerce orders placed on the web last year in two days!"

How SQL Server 2000 Compares...



Related Sessions

- 2-201: SQL Server 2000 Relational Engine Technology Enhancements
- 1-305: Storage Engine & Utilities Technology Enhancements
- 1-307: SQL Server 2000 Index Enhancements
- 2-314: Building High-Performance Systems with SQL Server 2000
- 2-315: Building High-Performance Applications with SQL Server 2000
- 2-311: Building "Scale Out" Applications with SQL Server 2000
- 1-310: SQL Server 2000 Tools for Performance Tuning

Questions And Feedback

POWER UP

Microsoft